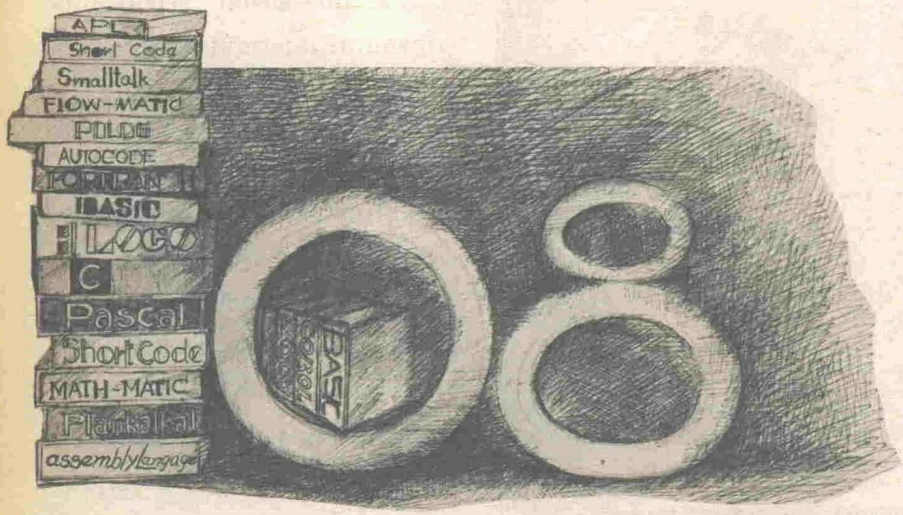
สหัส พรหมสิทธิ์

108 ภาษา ของคอมพิวเตอร์ (ตอน 1)



เสนอความเป็นมา
กำเนิดภาษาคอมพิวเตอร์
เพื่อเป็นสื่อเชื่อมโยง
ระหว่างคนกับคอมพิวเตอร์

ถ้าจะพูดว่า ภาษาเป็นเครื่องแสดงอารยธรรมของมนุษย์ก็น่าจะพูดต่อไปอย่างไม่ผิดได้อีกว่า ภาษาสำหรับคอมพิวเตอร์ก็เป็นเครื่องแสดงวัฒนธรรมการรุดหน้าในการใช้คอมพิวเตอร์ หรือใช้คอมพิวเตอร์ให้เป็นสื่อกลางไปกระตุ้นเตือนเครื่องมือจักรกลต่าง ๆ ให้เคลื่อนไหวทำงานอันจะเป็นประโยชน์ต่อมนุษย์ คอมพิวเตอร์นั้นตามลำพังตัวของมันเองก็น่าจะมีอานุภาพอยู่แล้ว แต่ครั้นเมื่อมันได้มีโอกาสไปทำงานเชื่อมโยงเพื่อสั่งการบังคับอุปกรณ์จักรกลอื่น ๆ ก็จะทำให้ทุกสิ่งทุกอย่างมีอานุภาพขึ้นไปอีก ในแง่ของการสามารถช่วยผ่อนแรงแบ่งเบาภาระจากมนุษย์ชนิดที่ง่ายที่สุด จนถึงขั้นที่จัดว่ายากมาก ๆ ได้อย่างสบาย ดังเช่นเรื่องราวของอิทธิพลของภาษาคอมพิวเตอร์ที่จะเล่าต่อไป

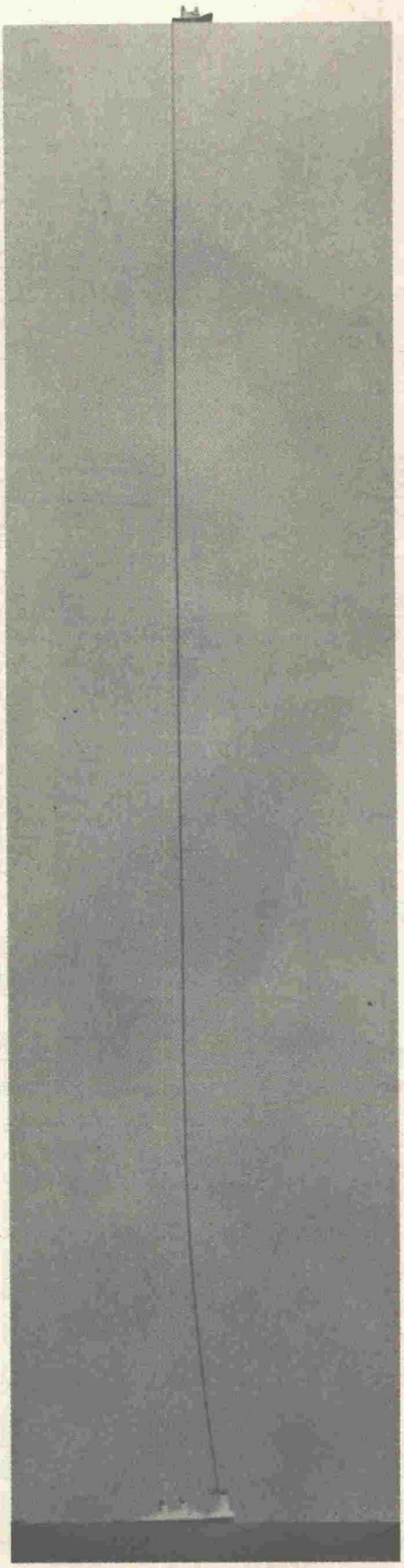
สองไมล์ในทะเลลึก

วิธีที่จะแลเห็นความสำคัญของสิ่งใดสิ่งหนึ่งนั้น น่าจะทำได้ง่ายที่สุดด้วยการยกอุทาหรณ์ของจริงของสิ่งนั้น ๆ ว่า ในโลกแห่งความเป็นจริงนั้นเขานำเทคโนโลยีต่าง ๆ รวมทั้งภาษาคอมพิวเตอร์ไปแก้ปัญหาใหญ่ ๆ ที่คนส่วนมากนึกไม่ถึงนั้นทำกันอย่างไร โดยจะพาผู้อ่านดำดิ่งลงสู่ก้นทะเลลึกอันเป็นจุดที่เรือเดินสมุทรที่ชื่อ ไตตานิค (Titanic) จมอยู่เมื่อเกือบ 80 ปีมาแล้ว เพื่อไปสำรวจสภาพของเรือลำนี้ การสำรวจนี้ทำเมื่อฤดูร้อนของปี 1985 ที่มหาสมุทรแอตแลนติคเหนือ โดยใช้ยานสำรวจใต้น้ำชื่อ อาร์โก้ (Argo) ที่มีความยาว 16 ฟุต ภายในยานสำรวจทะเลลึกที่มีความลึกมากกว่าสองไมล์นี้จะบรรจุอุปกรณ์เครื่องวัดต่าง ๆ เต็มไปหมด เช่น แสงไฟที่ให้ความสว่างสูงมาก ๆ โซน่าร์ วิดีโอ ฯลฯ ตรงส่วนบนผิวน้ำของห้วงทะเลแห่งนี้ก็มีเรือสำรวจขนาดใหญ่ชื่อ นอร์ (Knorr) จอดลอยลำเพื่อทำการติดต่อกับยานสำรวจอาร์โก้ตลอดเวลา บนเรือลำนี้จะมีนักสำรวจชาวอเมริกันและฝรั่งเศสร่วมกันทำงาน ซึ่งนักสำรวจส่วนใหญ่จะเพ่งมองไปที่จอภาพวิดีโอ อันเป็นเครื่องรับเพื่อดูภาพอันเกิดจากการถ่ายทอดขึ้นมาจากท้องทะเลลึกด้วยฝีมือของอาร์โก้ และดูว่าในขณะใดขณะหนึ่งนั้นอาร์โก้จะอยู่ในสภาพเช่นใดหรือกำลังทำอะไรอยู่

การค้นหาเรือไตตานิคนี้กระทำอยู่นานถึง 16 วัน จึงได้พบร่องรอยของการกระจัดกระจายของโลหะเป็นสิ่งที่แสดงตำบลที่ตั้งของเรือที่จมอยู่ รวมทั้งหม้อน้ำและซากศพของคนเป็นจำนวน 1,500 คน จากผู้โดยสารทั้งหมดในครั้งนั้นที่มีมากถึง 2,200 คน ที่ยังคงสิงสถิตย์อยู่

ในก้นทะเล ซึ่งการพยายามค้นหาทั้งหมดนี้เป็นการนำเทคโนโลยีด้านคอมพิวเตอร์แบบใหม่สุดมาช่วยอย่างเต็มที่ รวมทั้งภาษาคอมพิวเตอร์ต่าง ๆ ที่มีบทบาทไม่น้อยในการสำรวจใต้ทะเลในครั้งนี้ ในแง่ที่ว่า การทำงานของโซน่าร์ต่าง ๆ ตลอดจนการส่องแสงไฟและการถ่ายภาพใต้น้ำนี้เป็นผลมาจากการบังคับควบคุมของคอมพิวเตอร์กลุ่มหนึ่ง ที่ได้รับการโปรแกรมด้วยภาษาต่าง ๆ เพื่อให้เหมาะกับงานในแต่ละส่วน เช่น คอมพิวเตอร์ที่อยู่ในยานสำรวจอาร์โก้ซึ่งปราศจากคนก็จะได้รับการโปรแกรมด้วยภาษาที่เรียกว่า ฟอร์ธ (FORTH) อันเป็นภาษาที่มีถ้อยคำสั้นแต่มีความคล่องตัวสูง ที่ได้รับการพัฒนาขึ้นมาในตอนแรก ๆ เพื่อใช้ควบคุมการเคลื่อนไหวของกล้องโทรทรรศน์ดูดาว ตลอดจนสามารถใช้ควบคุมการทำงานของอุปกรณ์อื่น ๆ รวมทั้งควบคุมกระบวนการต่าง ๆ นับตั้งแต่การมอนิเตอร์การทำงานของหัวใจของมนุษย์ จนถึงการทำงานของกล้องถ่ายวิดีโอชนิดพิเศษ

ส่วนคอมพิวเตอร์บนเรือลาดตระเวณบนผิวน้ำที่ชื่อ นอร์ นั้นได้รับการโปรแกรมด้วยภาษาซีที่มีอานุภาพสูง แต่ลึกลับเป็นปริศนาเหมือนภาษาขอม เพียงเพราะเหตุผลอันสำคัญว่า เป็นภาษาคอมพิวเตอร์ที่กระชับและแม่นยำ เหมาะสำหรับการทำงานของคอมพิวเตอร์ในกรณีนี้ ส่วนระบบเทเลเมตตรี้ (Telemetry) ที่อยู่ตรงปลายทั้งสองของสายเคเบิลแบบ coaxial ที่มีเส้นผ่านศูนย์กลางเท่านิ้วคนนั้น จะทำหน้าที่เชื่อมโยงให้คอมพิวเตอร์บนเรือลาดตระเวณ (นอร์) สามารถติดต่อสื่อความกับยานสำรวจ (อาร์โก้) ได้อย่างมีประสิทธิผลตลอดเวลา โดยได้รับการโปรแกรมด้วยภาษาแอสเซมบลี (assembly) ที่ทรงประสิทธิภาพ จนทำให้สามารถเห็นสภาพการจมของเรือที่อยู่ลึกถึง 13,000 ฟุตได้ชนิดไม่เคยปรากฏมาก่อน ด้วยการใช้ภาษาคอมพิวเตอร์เพียงสามภาษาและใช้เทคโนโลยีใหม่สุดเข้าช่วย แต่ความจริงนั้นภาษาคอมพิวเตอร์มีนับเป็นร้อยอย่าง โดยที่แต่ละภาษาก็มีผู้คิดและพัฒนาขึ้นมาด้วยจุดมุ่งหมายต่าง ๆ กัน อันเป็นที่มาของคำว่า **"ร้อยแปดภาษาของคอมพิวเตอร์"** ซึ่งเป็นชื่อของบทความนี้



รูปที่ 1 ภาษาโปรแกรมต่าง ๆ (C, FORTH และ assembly) ที่เข้ามาช่วยให้ชาวโลกได้มีโอกาสเห็นสภาพการจมของเรือไตตานิคเป็นครั้งแรก เมื่อปี 1985 ที่อยู่ลึกถึง 13,000 ฟุต โดยไม่ต้องอาศัยคนดำน้ำลงไปแต่อย่างใด เพราะได้มีการใช้ระบบอัตโนมัติของยานสำรวจใต้น้ำเข้าช่วย

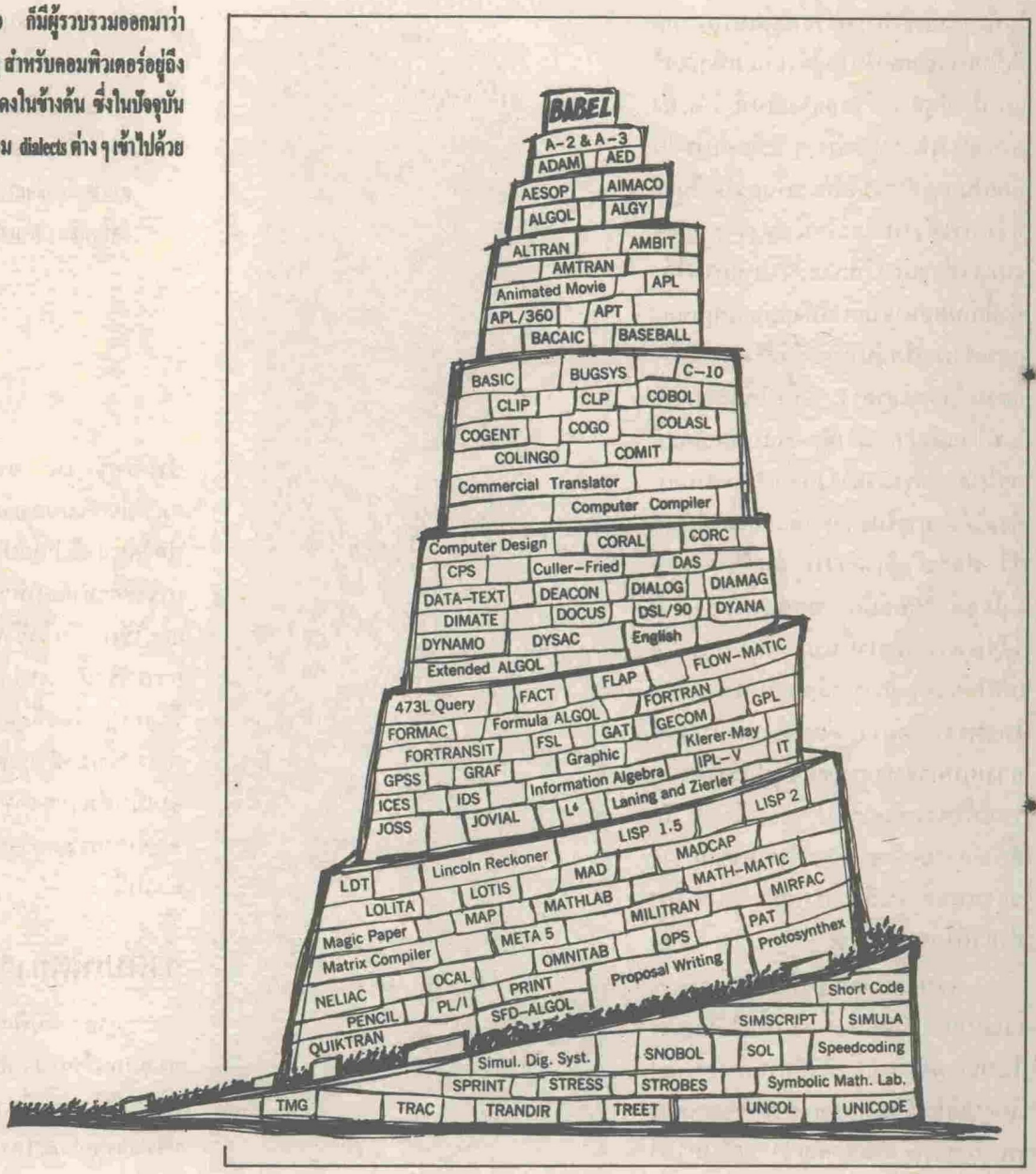
ลำดับเหตุการณ์

ภาษาคอมพิวเตอร์ส่วนใหญ่นั้นจะต่างจากผลิตภัณฑ์สินค้าหรือสิ่งประดิษฐ์ทั่ว ๆ ไป ตรงที่ว่ามักจะมีความคงทนท้าทายต่อกาลเวลา จึงจะขอเริ่มที่ภาษาอันแรก ๆ สุด ตามที่ประวัติบันทึกไว้แล้วตามด้วยภาษาอื่นที่เป็นที่รู้จักกันดีและกำกับด้วยปีที่ภาษานั้น ๆ เริ่มมีคน ส่วนใหญ่ใช้เพื่อให้ผู้อ่านเห็นภาพคร่าว ๆ ทั้งหมดเสียก่อนในตอนแรก เสร็จแล้วจึงจะพูดรายละเอียดในภายหลัง ดังนั้นถ้าจะลำดับเหตุการณ์อย่างย่นย่อที่สุดในเรื่องของภาษาคอมพิวเตอร์ที่เด่น ๆ ก็จะเป็นดังนี้

1946 Plankalkül

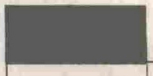
1949 Short Code

รูปที่ 2 เฉพาะแต่ในปี 1969 ก็มีผู้รวบรวมออกมาว่าได้เกิดมีภาษาต่าง ๆ สำหรับคอมพิวเตอร์อยู่ถึง 117 ภาษาแล้วดังแสดงในข้างต้น ซึ่งในปัจจุบันจะมีนับพัน ถ้าจะรวม dialects ต่าง ๆ เข้าไปด้วย



1950 EDSAC-assembly-language
1951 A-0
1952 AUTOCODE
1953 Speedcoding
1955 A-2
1956 FLOW-MATIC, IPL-1
1957 MATH-MATIC, FORTRAN
1958 ALGOL 58
1959 APT
1960 LISP, COBOL, ALGOL 60, JOVIAL
1964 PL/1, BASIC
1965 ALGOL-W
1967 Logo
1968 ALGOL 68
1969 APL
1970 Pascal
1971 FORTH
1972 Prolog, C
1979 Ada
1980 Smalltalk
1981 Modula-2

อันจะทำให้ผู้อ่านเข้าถึงประวัติความเป็นมาอย่างรวดเร็วว่า นับตั้งแต่ปี 1946 จนถึง 1981 เกิดอะไรขึ้นบ้างในวงการภาษาโปรแกรม ในเวลาเดียวกันก็ได้นำภาษาทั้งหมดที่มีผู้คิดค้นในประเทศต่าง ๆ มาแสดงไว้ใน รูปที่ 2 โดยสิ้นสุด



```
menge datum = tupel (ganz tag, wort monat, ganz Jahr);
funktion naechster tag nach = (datum x) datum:
                    wenn tag von x < monatslaenge (monat von x,  jahr von x)
                    dann      (tag von x + 1,  monat von x,  jahr von x)
                    wennaber  monat von x = "Dezember"
                    dann      (1,  "Januar",  jahr von x + 1)
                    ansonsten (1,  nachfolgemonat (monat von x),  jahr von x)
                    endewenn;
```

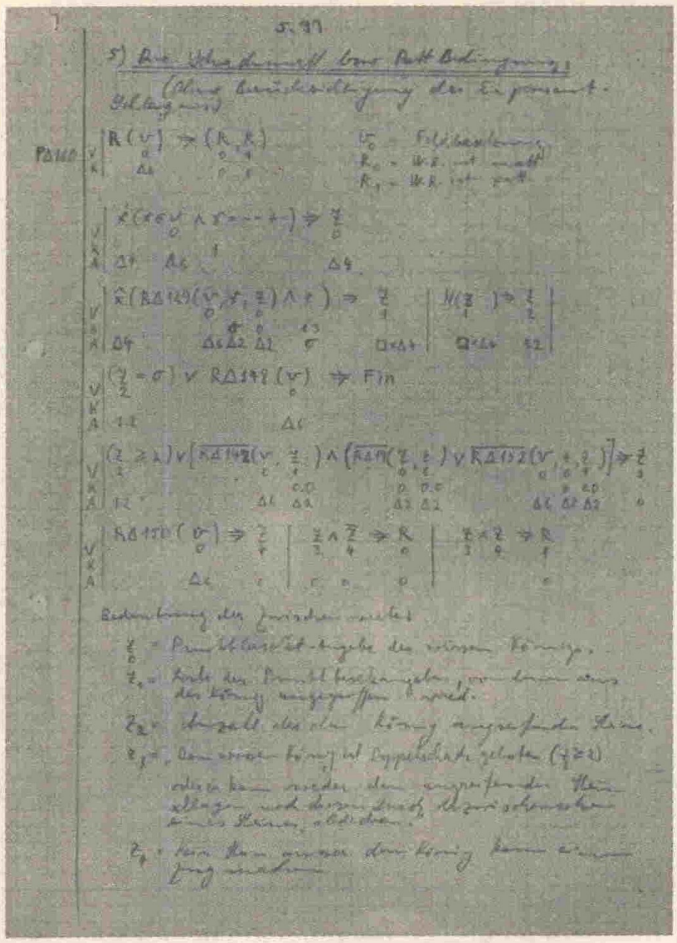
รูปที่ 3 โปรแกรมสำหรับคำนวณหา "วัน" ภายหลังจากที่บอก วันที่ เดือนและปี เข้าไปเป็นข้อมูลที่อยู่ในรูปของภาษา ALGOL 68 ซึ่งเมื่อนำโปรแกรมอันนี้ไปเขียนในรูปของภาษาเยอรมัน จีน และอเมริกันแล้วก็จะปรากฏดังภาพบน กลาง และล่างตามลำดับ โปรแกรมเมอร์ชาวเยอรมันและอเมริกันสามารถป้อนโปรแกรมของตนเข้าคอมพิวเตอร์ได้โดยตรงโดยอาศัย Translation Table เป็นสื่อ ส่วนของจีนนั้นต้องแปลงเป็นภาษาอังกฤษก่อนที่จะป้อนเข้าคอมพิวเตอร์

```
集合  日期 = 结构(整形日, 字符形月, 整形年);
函数  下一天 = (日期x)日期:
        如果  日对于x < 月的长度(月对于x, 年对于x)
        那么  (日对于x + 1, 月对于x, 年对于x)
        否则如果  月对于x = "12月份"
        那么  (1, "1月份", 年对于x + 1)
        否则  (1, 下一个月(月对于x, 年对于x)
        条件结束,
```

```
mode date = struct (int day,  string month, int year);
proc the day following  = (date x) date:
                    if day of x < length of month (month of x,  year of x)
                    then (day of x + 1, month of x,  year of x)
                    elif  month of x = "December"
                    then (1, "January",  year of x + 1)
                    else (1,  successor of month (month of x),  year of x)
                    fi;
```

แค่เพียงปี 1969 เพื่อให้ผู้อ่านเห็นอะไร ๆ ละเอียดขึ้นไปอีกว่า ภาษาคอมพิวเตอร์นั้นทำไมจึงมากมายเพียงนี้ อันนี้ก็เป็นเพียงอารัมภบทแบบรวบรัดที่สุดก่อนวกเข้ารายละเอียดอันเป็นเสมือนประวัติศาสตร์ เมื่อพูดถึงภาษาคอมพิวเตอร์ ก็ต้องพูดถึงฮาร์ดแวร์ของคอมพิวเตอร์บ้างเพื่อเป็นภาพสะท้อนว่า วิวัฒนาการของภาษานั้นมีส่วนผูกพันกับฮาร์ดแวร์อย่างไร เช่น คอมพิวเตอร์ซึ่งใช้งานได้ทั่ว ๆ ไปที่สามารถโปรแกรมเข้าไปได้เป็นอันแรกของโลกที่ชื่อ Mark 1 นั้น จะมีขนาดใหญ่เทอะทะมาก และทำงานโดยใช้รีเลย์ ชาฟต์ เฟืองและสิ่งหมุนต่าง ๆ จึงทำให้คอมพิวเตอร์ที่สร้างระหว่างสงครามโลกครั้งที่สองนี้ยาวถึง 51 ฟุต ส่วนการป้อนโปรแกรมที่เป็นคำสั่งต่าง ๆ นั้นได้อาศัยเทปกระดาษ ต่อมาก็มี ENIAC ที่สร้างขึ้นเมื่อปี 1945 ที่มหาวิทยาลัยเพ็นซิลเวเนียที่ทันสมัยกว่า Mark 1 ที่ทำงานด้วยระบบกลและไฟฟ้า ซึ่งเป็นของฮาร์วาร์ด ENIAC นั้นเป็นอิเล็กทรอนิกส์ล้วน ๆ แต่ก็ยังโปรแกรมยากเหมือน Mark 1 และไม่ยอมรับการป้อนโปรแกรมทางเทปกระดาษ อันทำให้เกิดความไม่สะดวกมากขึ้นไปอีก เพราะการพัฒนา ENIAC นั้นเกิดจากความจำเป็นที่ต้องทำอย่างเร่งรีบในยามสงคราม การนึกถึงเรื่องของโปรแกรมจึงมาทีหลังเรื่องของฮาร์ดแวร์ การป้อนคำสั่งในกรณีของ ENIAC นี้ต้องใช้ทั้งการตั้งสวิตช์นับเป็นพัน ๆ อัน พร้อมทั้งการต่อโยงสายเคเบิลนับร้อยเข้าไปใน plug-boards ต่าง ๆ จนทำให้ดูซับซ้อนยุ่งเหยิงเอามาก ๆ

คอมพิวเตอร์ในยุคแรก ๆ เช่น Mark 1 และ ENIAC ได้ให้บทเรียนไว้หลาย ๆ อย่าง โดยเฉพาะในแง่ที่ต้องพยายามหาวิธีการจะพูดจาสื่อความกับมันด้วยวิธีที่สะดวกที่สุดเท่าที่จะทำได้ เพราะจะเป็นวิธีเดียวเท่านั้นที่ใช้ศักยภาพของ



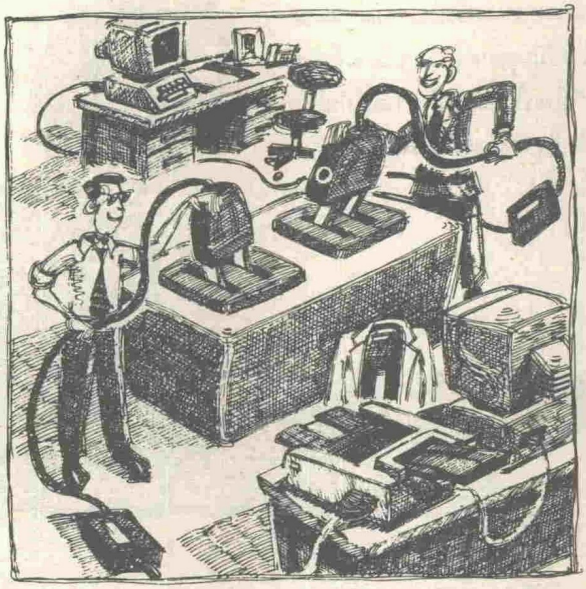
รูปที่ 4 ภาษาคอมพิวเตอร์ที่เรียกว่า Plankalkül ที่คิดขึ้นมาโดยวิศวกรชาวเยอรมันผู้มีนามว่า Konrad Zuse ในภาพที่แสดงไว้นี้เป็นส่วนหนึ่งของโปรแกรมเล่นหมากรุก โดยได้แสดง steps ต่าง ๆ ไว้ เช่น white king จะผจญกับปัญหาได้ด้วยวิธีใดบ้าง เป็นที่ยอมรับในปัจจุบันว่าภาษา Plankalkül นี้น่าจะเป็นภาษาขั้นสูง (high level language) อันแรกของโลก

เครื่องจักรเช่นนี้ให้เกิดประโยชน์สูงสุดได้ ซึ่งในขณะที่ ENIAC กำลังทำงานอยู่นั้น ก็เป็นไปได้ว่ากำลังมีผู้คิดค้นภาษาขั้นสูงที่เรียกว่า high level computer language อยู่ในอีกมุมหนึ่งของโลกอยู่ อันเนื่องมาจากนิสัยในการเล็งการณ์ไกลของมนุษย์ เช่น กรณีของ Konrad Zuse (คอนแรด ซู๊ส) วิศวกรชาวเยอรมัน

Plankalkül ภาษาชั้นสูงอันแรกของโลก

ภาษาคอมพิวเตอร์ชั้นสูงที่เป็นที่ยอมรับว่าเป็นอันแรกของโลกนี้พัฒนาขึ้นมาด้วยฝีมือของ นายคอนแรด ซู๊ส (Conrad Zuse) โดยมีชื่อว่า Plankalkül ซึ่งแปลว่า Plan Calculus ซู๊สเป็นวิศวกรเยอรมันที่เป็นนักออกแบบคอมพิวเตอร์และนักซอฟต์แวร์ที่เติบโตมาเป็นหนุ่มในช่วงสงครามโลกครั้งที่สอง โดยอาศัยอยู่ที่กรุงเบอร์ลินและต่อมาได้ลี้ภัยมาอยู่ที่ Bavarian Alps ภายหลังสงครามโลก บุคคลผู้นี้ได้ค้นคว้าในด้านคอมพิวเตอร์ไว้มากมาย ทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ ส่วน Plankalkül ได้มาพัฒนาสำเร็จเสร็จเป็นรูปเป็นร่างที่เมืองบาวาเรียนี่เอง ผลงานอันมากมายของซู๊สนั้นเพิ่งจะมาปรากฏตีพิมพ์ในปี 1972 อันแสดงอัจฉริยภาพของบุคคลผู้นี้อย่างมาก ๆ ในแง่ของการพัฒนาแนวความคิดในการเขียนโปรแกรมที่จัดว่าล้ำยุคทันสมัยจนทำให้สงสัยกันว่า ถ้าโลกได้มีโอกาสล่วงรู้ผลงานของซู๊สเร็วกว่านี้อีก 10 ปี ป่านนี้โลกของคอมพิวเตอร์คงจะโสภากว่านี้อีกหลายเท่า ซึ่งผู้เขียนได้นำอักขระของซู๊สมาแสดงเป็นตัวอย่างไว้ในรูปที่ 4 อันเป็นภาพที่ชัดที่สุดที่จะถ่ายสำเนามาให้เห็นได้

ในขณะที่ซู๊สขะมักเขม้นพัฒนาภาษาคอมพิวเตอร์อยู่นั้น ทั้งสหรัฐฯและอังกฤษก็ไม่ยอมเลิกราที่จะผจญอุปสรรคในด้านภาษาโปรแกรมที่จะให้ใช้ได้ง่ายและดีขึ้น แต่ความก้าวหน้าในขณะนั้นก็เชื่องช้าเหลือเกิน เพราะเป็นกรรมอยู่อย่างว่า คอมพิวเตอร์แต่ละเครื่องในยุคนั้น (ราวปี 1946-50) จะมีแมชีนโค้ดและวิธีโปรแกรมเป็นเอกเทศ คือของใครของมัน จนกระทั่งมีนักโปรแกรมผู้หนึ่งที่ชื่อ Grace Murray Hopper ซึ่งเป็นชาวอเมริกันที่ได้ทำงานร่วมกับนักคอมพิวเตอร์ชั้นนำอื่น ๆ ในขณะนั้น (ราวปี 1949) ได้ร่วมกันพัฒนาภาษาใหม่ที่เรียกว่า Short Code เป็นผลสำเร็จในรูปของการใช้งานที่เป็นอิเล็กทรอนิกส์ดิกชันนารี และจัดเป็นภาษาชั้นสูงแบบโบราณที่ดีกว่าแมชีนโค้ดหลายเท่า และก็เป็นภาษาที่จะเป็นพื้นฐานเพื่อให้มีการ



พัฒนาภาษาคอมพิวเตอร์อื่น ๆ ที่จะตามมาในภายหลัง เพราะ Short Code นี้เป็นภาษาที่เริ่มปลดแอกปลีกตัวเองจากแมชีนโค้ดในแง่ที่โปรแกรมเมอร์ไม่ต้องเขียนโปรแกรมเป็นแมชีนโค้ดอันแสนยากอีกต่อไป

ยุคต่อมา (ปี 1950-60) และอุบัติการณ์ของคอมพายเลอร์

ในช่วงสงครามและหลังสงครามโลกครั้งที่สองนั้นสหรัฐฯก็ได้มีคอมพิวเตอร์ เช่น ENIAC ในเวลาอันไล่เลี่ยกันกับที่ทางอังกฤษก็มีคอมพิวเตอร์ที่ชื่อ EDSAC ที่ใช้การได้เมื่อปี 1949 ซึ่งยังมีประสิทธิภาพต่ำอยู่มาก คือมีอะไรเข้ามากระทบนิด ๆ หน่อย ๆ ก็ใช้การไม่ได้ต้องหยุดซ่อม โดยในระยะแรก ๆ นั้น EDSAC ทำหน้าที่ทางคำนวณที่เรียกว่า basic operations ได้เพียง 18 อย่างเท่านั้น ส่วนการใช้คำสั่งก็จะอยู่ในรูปของ mnemonics ต่าง ๆ เช่น S หมายถึง ลบ, I หมายถึง การอ่านข้อมูล, T หมายถึง การโอนข้อมูลไปเข้า storage, Z หมายถึงให้คอมพิวเตอร์หยุดทำงาน ซึ่งตอนนั้นก็เริ่มมีคีย์บอร์ดให้โปรแกรมเมอร์ใช้ป้อนคำสั่งเข้าเครื่องได้แล้ว

จากภาษาแบบ mnemonics นี้ก็ได้มีการพัฒนาต่อมา จนกลายเป็นภาษาแอสเซมบลีที่เป็น mnemonics อย่างหนึ่งที่มีทั้งส่วนที่เป็นสัญลักษณ์คือตัว mnemonics และส่วนที่เป็นรูทีนย่อย ๆ ที่ต้องใช้บ่อย ๆ ซึ่งภาษานี้ก็มีใช้กันจนปัจจุบัน เพราะเป็นภาษาที่ใกล้ชิดสนิทแนบกับภาษาเครื่องอย่างมาก ๆ ในแง่ของการสามารถใช้เนื้อที่ในเมมโมรีของคอมพิวเตอร์ได้อย่างประหยัดที่สุด และวิ่งได้เร็วที่สุดขณะทำการคำนวณ ซึ่งผู้อ่านคงพอจะจำเรื่องราวของการสำรวจเรือไตตานิคในตอนต้น ๆ ที่ได้มีการใช้ภาษาแอสเซมบลีเข้าช่วยด้วย เพราะความมีประสิทธิภาพในบางแง่บางมุมของภาษาชนิดนี้

จากแอสเซมบลีก็มีภาษาที่เรียกว่า AUTOCODE (ปี 1952) ซึ่งไม่ค่อยได้รับความนิยม ภาษานี้พัฒนาขึ้นมาด้วยฝีมือของนาย Glennic ที่มหาวิทยาลัยแมนเชสเตอร์ในอังกฤษ การอับรัศมีของ AUTOCODE ซึ่งเนื่องมาจากความไม่นิยมของโปรแกรมเมอร์ในยุคนั้นที่เห็นว่า ฮาร์ดแวร์ยังมีขีดจำกัดในด้านความจุและความเร็วอย่างมาก ๆ อยู่ อันยังคงจะเหมาะที่สุดที่จะเขียนโปรแกรมด้วยแมชีนโค้ด ที่จะทำให้ได้เนื้องานมากและวิ่งได้เร็วในเวลาคำนวณ อีกเรื่องก็คือ แมชีนโค้ดนั้นรู้อยู่เฉพาะในแวดวงของมืออาชีพที่ใครจะมาแย่งงานไปทำไม่ได้ง่าย ๆ

ในช่วงนี้เองที่นักลงทุนในสหรัฐฯ และอังกฤษ เริ่มเห็นความสำคัญของอุตสาหกรรมคอมพิวเตอร์ว่านับแต่จะต้องมีความสำคัญขึ้นเรื่อย ๆ เมื่อวันเวลาผ่านไป ซึ่งในสหรัฐฯนั้นทั้งไอบีเอ็มและ Remington Rand ก็ได้ไหวตัวในเรื่องนี้มานานแล้วนับตั้งแต่สมัยสงครามโลก เช่น ในปี 1951 ที่ Remington Rand ได้ส่งมอบคอมพิวเตอร์ทางการค้าเครื่องแรกที่ชื่อ UNIVAC ให้สำนักงานสำรวจสำมะโนประชากรของสหรัฐฯ ซึ่งตอนนี้สตรีผู้มีนามว่า Grace Murray Hopper ที่ได้เคยเกริ่นแต่แรกทั้งในเรื่องของโปรแกรมและซอฟต์แวร์นั้นก็ยังคงทำงานในสายนี้มาเรื่อย ๆ ด้วยความกระตือรือร้น ในฐานะที่เธอเป็นทั้งนักคณิตศาสตร์ (Ph.D คณิตศาสตร์จากเยลเมื่อปี 1928) และนักวิศวกรคอมพิวเตอร์ที่รู้อะไร ๆ ดีมาก ๆ คนหนึ่งของยุค เคยรับราชการในกองทัพเรือของสหรัฐฯในช่วงสงคราม และในปี 1949 ก็ลาออกมาทำงานกับบริษัท Eckert-Mauchly Computer Corporation ซึ่งต่อมาบริษัทนี้ก็ถูก Remington Rand ซื้อกิจการไปรวมอยู่ในเครือ แต่ก่อนที่จะถูกซื้อกิจการนั้น บริษัทของ Eckert กับ Mauchly นี้ได้เป็นผู้สร้างคอมพิวเตอร์ที่ชื่อว่า Universal Automatic Computer หรือ

รูปที่ 5 โฉมหน้าของดร.เกรซ ฮ็อปเปอร์ นักโปรแกรมเมอร์มือฉมังในยุคแรก ๆ ของการมีคอมพิวเตอร์ และก็ยังยิ่งยงในเรื่องฮาร์ดแวร์และโปรแกรม มาจนถึงปัจจุบันที่เธอยังมีชีวิตอยู่

UNIVAC ขึ้นมาเองเมื่อราวปี 1949 อันเป็นผลที่ Eckert และ Mauchly ยกทีมกันลาออกจากมหาวิทยาลัยเพ็นซิลเวเนีย (เพราะทะเลาะกันในเรื่องของสิทธิประโยชน์ของอิเล็กทรอนิกส์คอมพิวเตอร์) มาตั้งบริษัท ซึ่งพอดีกับที่ดร.เกรซ ฮ็อปเปอร์ลาออกจากกองทัพเรือจึงได้มาทำงานกับบริษัทนี้ ซึ่งผู้อ่านจะเห็นโฉมหน้าของสตรีผู้นี้ได้จากรูปที่ 5 อันเป็นภาพขณะที่เธอยังทำงานอยู่กองทัพเรือสหรัฐฯ โดยเริ่มงานที่กองทัพนี้ในด้านคอมพิวเตอร์

จะขอวกกลับมาพูดถึงเหตุการณ์หลังจากที่ Remington Rand ได้ส่งมอบ UNIVAC เครื่องแรกให้สำนักงานสำรวจสำมะโนประชากรของสหรัฐฯในปี 1951 ซึ่งตอนนี้ ดร.ฮ็อปเปอร์ก็มาอยู่ในสังกัดของ Rand เรียบร้อยแล้วอันเนื่องมาจากการรวมกิจการ ในปีเดียวกันนี้ดร.ฮ็อปเปอร์ก็ได้รับมอบหมายให้สร้างไลบรารีสำหรับรูทีนย่อยทางคณิตศาสตร์ที่เป็นมาตรฐานและใช้กันบ่อย ๆ ไว้ใช้กับ UNIVAC อันเป็นเวลาเดียวกับที่ Wilkes และโปรแกรมเมอร์อีกสองคนที่มหาวิทยาลัยเคมบริดจ์ได้พิมพ์หนังสือเล่มแรกของโลกที่เกี่ยวแก่การเขียนโปรแกรมสำหรับคอมพิวเตอร์ EDSAC ที่สร้างที่มหาวิทยาลัยนี้ ซึ่งเนื้อหาสาระของหนังสือเล่มนี้ก็จะเกี่ยวข้องกับการเขียนโปรแกรมด้วยภาษาแอสเซมบลี ที่พวกโปรแกรมเมอร์ในประเทศอื่น ๆ สนใจกันมาก เมื่อดร.ฮ็อปเปอร์ทราบข่าวนี้ก็นึกถึงช่องทางใหม่ ๆ ที่จะมาในมาดที่เหนือกว่าการใช้แค่เพียงภาษาแอสเซมบลีมาสื่อความกับคอมพิวเตอร์ ตามวิสัยของคนอเมริกันที่รักความสะดวก ซึ่งการรักความสะดวกนี้จะเป็นผลบุญในเวลาต่อมาสำหรับผู้ชอบเขียนโปรแกรมที่กลัวความลำบาก นั่นคือ ดร.ฮ็อปเปอร์ได้เล็งเห็นหนทางว่าจะสามารถใช้ประสบการณ์ที่เธอมีในภาษา Short Code มาออกแบบ "ระบบ" อันหนึ่งที่จะสามารถแปลโปรแกรมที่เขียนด้วยภาษาขั้นสูง (high level language) ให้เป็นภาษาเครื่อง ซึ่งหน้าที่ของ "ระบบ" หรือ Translating โปรแกรมนี้จะใช้เวลาเพียงไม่กี่วินาทีให้คอมพิวเตอร์จัดการกับการใช้รูทีนย่อยต่าง ๆ ตลอดจนการจัดสรรการใช้หน่วยความจำของคอมพิวเตอร์ และแปลภาษาขั้นสูงให้เป็นแมชีนโค้ด อันเป็นที่มาของคำว่าคอมพายเลอร์ (Compiler) และเป็นคอมพายเลอร์อันแรกที่เป็นที่รู้จักกันดีมากกว่าคอมพายเลอร์ที่เรียกว่า AUTO-CODE ซึ่งการทำงานด้วยการแปลงภาษาขั้นสูงให้เป็นแมชีนโค้ดคราวละเป็นกะบิ ๆ เช่นคอมพายเลอร์อันใหม่นี้จะแตกต่างจากการทำงานของ AUTO-CODE ที่แปลทีละบรรทัดในรูปของอินเตอร์พรีเตอร์เป็นอย่างมากดังที่รู้ ๆ กันอยู่

ดร.ฮ็อปเปอร์ได้ให้ชื่อคอมพายเลอร์อันแรกที่เธอและโปรแกรมเมอร์กลุ่มหนึ่งเป็นผู้สร้างมันขึ้นมาว่า A-0 และได้รับการปรับปรุงในเวลาต่อมาให้เป็น A-1, A-2 และ A-3 จนกระทั่งมีชื่อทางการค้าว่า MATH-MATIC ในที่สุดในช่วงนี้เอง (ปี 1953) ที่ไอบีเอ็มได้เข็นอิเล็กทรอนิกส์คอมพิวเตอร์เครื่องแรก (รุ่น 701) ของตนออกสู่ตลาด พร้อมด้วยภาษาแอสเซมบลีที่เรียกว่า Speed coding ในขณะเดียวกันที่สถาบันเอ็มไอทีก็มีการค้นคว้าเพื่อสร้างคอมพายเลอร์ที่จะมาใช้กับคอมพิวเตอร์ที่เรียกว่า Whirlwind ทำให้การเขียนโปรแกรมง่ายขึ้นมากจนเกือบจะเหมือนกับภาษาฟอร์แทรน ส่วนทาง ดร.ฮ็อปเปอร์ก็ยังคงมีไฟแรงเหมือนเดิม คือเธอได้คิดหาทางที่จะสื่อความกับคอมพิวเตอร์ให้ได้สะดวกมากขึ้นไปอีก อันจะเกิดผลดีในทางการค้าให้กับ

รูปที่ 6 นักคณิตศาสตร์ผู้มีนามว่า John Backus ที่เข้าทำงานกับไอบีเอ็มเมื่ออายุ 28 และเป็นผู้นำทีมคนหนุ่มนักเขียนโปรแกรมของบริษัทนี้ได้สร้างฟอร์แทรนจนสำเร็จเมื่อปี 1957 โดยใช้เวลาพัฒนาอยู่ 3 ปีเพื่อให้ได้มาซึ่งภาษาชั้นสูงอย่างแท้จริง

Remington ในระยะแรก ๆ ผู้บริหารของ Remington ไม่เชื่อว่าคอมพิวเตอร์จะเข้าใจภาษาอังกฤษที่ป้อนเข้าไปโดยตรงจากโปรแกรมเมอร์ แต่เมื่อดร.ฮ็อปเปอร์หว่านล้อมด้วยการสาธิตของจริงด้วยภาษาอังกฤษ ฝรั่งเศส และเยอรมัน ให้ดูว่าคอมพิวเตอร์นั้นรับภาษาเหล่านี้ได้ หลายคนจึงเชื่อใจในฝีมือว่าเธอจะทำเรื่องนี้ได้สำเร็จ จนกลายมาเป็นคอมพายเลอร์ที่ชื่อ FLOW-MATIC อันนี้ก็จะเป็นเวลาราวปี 1955 ที่ได้มีการวางรากฐานสำคัญ ๆ ในด้านคอมพิวเตอร์ไว้มากพอที่จะใช้เป็นย่างก้าวสำหรับยุคทองของการพัฒนาภาษาอื่น ๆ ต่อไป

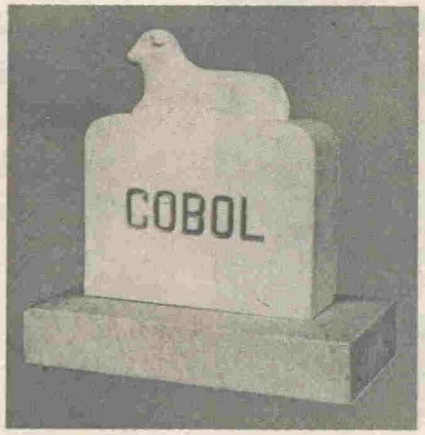
สามภาษาผู้ยิ่งใหญ่ ฟอร์แทรน โคบอล อัลกอล

ภาษาคอมพิวเตอร์ระดับสูงที่ใช้ง่ายจนจัดเป็นภาษาทางการค้านั้นว่ากันว่ามีอยู่สามภาษาในช่วงแรกนับจากปี 1957 เป็นต้นมาคือ ฟอร์แทรน โคบอล และอัลกอล จนน่าจะถือเป็นภาษาคลาสสิกได้ ภาษาฟอร์แทรนเป็นผลจากการค้นคว้าของไอบีเอ็มที่เริ่มขึ้นเมื่อปี 1954 โดยมีผู้นำทีมชื่อ John Backus และมาสำเร็จเมื่อปี 1957 รูปที่ 6 เป็นโฉมหน้าของ John Backus นี่ก็เป็นเรื่องย่อ ๆ เกี่ยวกับฟอร์แทรน เพราะมีความแปลกใหม่อื่น ๆ ให้พูดอีกมาก เช่น ความเป็นมาของโคบอลและอัลกอลที่น่าสนใจไม่น้อย

จะขอกลับมาพูดถึงผลงานของดร.ฮ็อปเปอร์ต่อไปอีก เพราะบุคคลผู้นี้ยังมีบทบาทอีกมากในแง่ของการเกิดมาเป็นนักคอมพิวเตอร์ภาคปฏิบัติอย่างแท้จริง ซึ่งในช่วงนี้ก็เริ่มมีภาษาคอมพิวเตอร์อื่น ๆ ที่ใช้ในทางธุรกิจการค้าปรากฏออกมาบ้างแล้ว เช่น FLOW-MATIC ของ Rand และ Commercial Translator ของไอบีเอ็ม รวมทั้งภาษาชื่อ AIMACO ของกองทัพอากาศสหรัฐฯ แต่ยังไม่เป็นที่ยอมรับของคนทั่วไป ส่วนภาษาฟอร์แทรนนั้นก็สร้างขึ้นมาใช้กับงานทางวิทยาศาสตร์และวิศวกรรมอันไม่เหมาะจะใช้ในทางธุรกิจ จนกระทั่งในปี 1959 จึงได้มีการประชุมกันที่มหาวิทยาลัยเพ็นซิลเวเนีย โดยมีผู้เข้าร่วมประชุมที่ประกอบไปด้วยนักอุตสาหกรรม นักธุรกิจ และบุคคลในวงการต่าง ๆ ที่เห็นว่าถึงเวลาแล้วที่ต้องรีบเร่งทำอะไรสักอย่างเพื่อแก้ปัญหาเรื่องภาษาคอมพิวเตอร์ที่เหมาะสมที่สุดที่จะใช้ในทางการค้า และภาษาเช่นนี้จะต้องไม่ผูกพันกับคอมพิวเตอร์ของบริษัทไหนเป็นการเฉพาะ คือจะต้องใช้ได้กับคอมพิวเตอร์ทุกแบบ ภาษาคอมพิวเตอร์ที่จะใช้ในทางธุรกิจการค้านั้นจะต้องต่างจากฟอร์แทรนที่มักไม่ต้องการการแก้สมการหรือการคำนวณอันซับซ้อน แต่สิ่งที่ธุรกิจการค้าโดยทั่วไปต้องการนั้นจะต้องเป็นภาษาคอมพิวเตอร์ที่เหมือนกับการใช้ภาษาอังกฤษมากที่สุด นอกเหนือจากการทำหน้าที่ที่จำเป็นอื่น ๆ ได้ เช่น sorting และ merging พวกไฟล์ข้อมูลต่าง ๆ รวมทั้งการทำรายงานออกมา ซึ่งทั้งหมดนี้จะต้องเป็นไปอย่างแคล่วคล่องรวดเร็ว ทั้งในด้านของการเขียนโปรแกรมและการประมวลผลของคอมพิวเตอร์ แต่ใครเล่าที่จะมาเป็นผู้ให้การอุปถัมภ์สำหรับการค้นคว้าและพัฒนาภาษาอันนี้ เพราะน่าจะเป็นงานใหญ่จนไม่มีบริษัทสร้างคอมพิวเตอร์แห่งไหนจะรับไปทำได้ในขณะนั้น ตอนนี้เองที่ดร.ฮ็อปเปอร์ได้เข้ามามีบทบาทอีกครั้งในฐานะผู้นำของตัวแทนจากกลุ่มต่าง ๆ ที่จะเข้าไปขอความสนับสนุนในการสร้างภาษาคอมพิวเตอร์สำหรับธุรกิจการค้าขนานแท้จากกระทรวงกลาโหมสหรัฐฯ ในฐานะที่กระทรวงนี้เป็นผู้หนึ่งที่ใช้คอมพิวเตอร์มากที่สุดแห่งหนึ่งของโลกในขณะนั้น

และเป็นหน่วยงานของรัฐที่มีสรรพกำลังในด้านต่าง ๆ ที่จะเป็นผู้นำในเรื่องนี้ได้ ซึ่งกระทรวงกลาโหมก็แลเห็นความสำคัญจึงตกลงให้การช่วยเหลือด้วยการจัดให้มีการประชุมในระดับ conference พิเศษที่ว่าด้วยภาษาคอมพิวเตอร์ขึ้นเป็นครั้งแรก เมื่อเดือนพฤษภาคม 1959 โดยมีชาร์ลส์ ฟิลลิปส์ ซึ่งเป็นผู้อำนวยการของหน่วยงานคาต้าซิสเต็มส์รีเสิร์ชในกระทรวงกลาโหมเป็นประธาน ซึ่งต่อมา conference อันนี้ได้กลายเป็น Conference on Data System Languages (CODASYL) ตัวแทนสำคัญกลุ่มหนึ่งที่ร่วมประชุมในขณะนั้นจะเป็นผู้สร้างคอมพิวเตอร์รายใหญ่ ๆ เช่น ฮันนี่เวลล์, ยีอี, เบอร์โรห์, เอ็นซีอาร์, ฟิลโก้, สเปอร์รี่ แรนด์ (ซึ่งแต่เดิมเป็น Remington Rand), อาร์ซีเอ, ซิลเวเนีย อิเล็กทริคโปรดักส์ และไอบีเอ็ม คราวนี้ก็มาดูถ้อยแถลงของนายชาร์ลส์ ฟิลลิปส์ ผู้ชอบพูดแบบขวานผ่าซากบ้าง ว่าเขาได้ลั่นวาจาอันเป็นการปิดประชุมครั้งแรกเมื่อสัปดาห์สุดท้ายของเดือนพฤษภาคม 1959 ไว้อย่างไร คำขอร้องแกมบังคับของนายฟิลลิปส์ใน conference ครั้งแรกนั้นมีเพียงสั้น ๆ ว่า **"ทางเพนตากอน (Pentagon) มีความต้องการภาษาโปรแกรมทางธุรกิจการค้าที่มีความสมบูรณ์ เป็นอันหนึ่งอันเดียวกันเป็นอย่างมาก และความต้องการนี้ค่อนข้างเร่งด่วน"**

หลังจากการประชุมครั้งแรกซึ่งต่อมาได้กลายเป็น CODASYL นั้น ก็มีการตั้งคณะกรรมการ คณะทำงานและคณะอะไรต่อมิอะไร เหมือนเมืองไทยในสมัยนี้ แต่ที่แน่ ๆ ก็คือการดำเนินในเรื่องการสร้างภาษาอันใหม่นี้ได้เป็นไปอย่างล่าช้าตามฟอร์ม จะขอเล่าสั้น ๆ ว่ามีอยู่คราวหนึ่ง ได้มีกรรมการชุดหนึ่ง



รูปที่ 7 ฮวงซุ้ยรูปปั้นแกะตัวนี้ถูกส่งไปให้นายชาร์ลส์ ฟิลลิปส์ ด้วยน้ำมือของนายโฮวาร์ด บรอมเบอร์ก จากอาร์ซีเอ เพราะทนดูความล่าช้าในระยะแรก ๆ ไม่ไหว จึงต้องส่งสิ่งเตือนใจไปให้ผู้เป็นประธาน เมื่อต้นปี 1960 โดยนายบรอมเบอร์กได้สารภาพการกระทำของเขาในปี 1985 ขณะมีการฉลองอายุครบ 25 ปีของการใช้ภาษาโคบอล

พยายามที่จะแก้ปัญหาในการสร้างภาษาใหม่โดยวิธีลัดด้วยการรวมภาษาที่มีอยู่แล้วคือ FLOW-MATIC, AIMACO และ Commercial Translator เข้าด้วยกัน แต่ทำไม่ได้ เพราะเป็นการฝืนทำในสิ่งที่เป็นไปได้ยาก จนในที่สุดคณะกรรมการชุดใหญ่สุดที่เรียกว่า CODASYL Executive Committee ซึ่งมีนายชาร์ลส์ ฟิลลิปส์ เป็นประธาน และมีหน้าที่ต้องติดตามดูแลการทำงานของคณะกรรมการอีกสามชุด (คือคณะกรรมการระยะสั้น ระยะกลาง และระยะยาวที่เป็นกลุ่มทำงาน) ก็อดรนทนไม่ได้จนต้องหาทางผลักดันให้กรรมการต่าง ๆ พยายามลืมของเก่าแล้วสร้างภาษาใหม่ขึ้นมาแบบเอกเทศ อันเป็นที่มาของคำว่า COBOL หรือ Common Business Oriented Language ซึ่งงานสำคัญอันหนึ่งก็คือการสร้างข้อกำหนดรายละเอียดสำหรับเรื่องนี้

การสร้างข้อกำหนดรายละเอียด (specification) สำหรับโคบอลยังคงเต็มไปด้วยข้อขัดแย้งในแง่ของผลประโยชน์ ความไม่เข้าใจต่าง ๆ อีกมากมาย แต่ในที่สุดก็ยุติได้ด้วยการยอมรับภาษาโคบอลอย่างเป็นทางการ หลังจากที่โต้เถียงกันมานานว่าจะใช้ภาษาอีกอันที่ชื่อว่า FACT (Fully Automatic Compiling Technique) ดีหรือไม่ ซึ่ง FACT ได้ก่อกำเนิดขึ้นมาด้วยฝีมือของคณะกรรมการระยะกลาง หรือที่เรียกว่า intermediate-range committee ซึ่งผู้อยู่เบื้องหลังที่เดินงานนี้อย่างเงียบ ๆ ก็คือบริษัทฮันนี่เวลล์นั่นเอง ในที่สุด FACT ก็ตกไปแม้จะเป็นภาษาที่เต็มไปด้วยนวกรรมใหม่ ๆ ในยุคนั้น เช่น การสามารถป้อนข้อมูลด้วยบัตร รวมทั้งมีการเขียนรายงานที่ใช้งานได้ง่าย แต่น้ำน้อยย่อมแพ้ไฟ โดยเฉพาะการค้านั้นย่อมไม่เข้าใครออกใคร ถ้าใครสักคนรู้ว่าอีกรายหนึ่งจะได้ดีกว่าตัว ดังนั้นเมื่อมีการประชุมคณะกรรมการในระดับใหญ่สุดเมื่อเดือนมกราคม 1960 ทุกคนก็ลงคะแนนให้โคบอลเป็นธรรมดา ซึ่งวงการธุรกิจการค้าต่าง ๆ ในสหรัฐฯ ในขณะนั้นก็ให้การยอมรับโคบอลตั้งแต่ต้นอยู่แล้ว

ข้อกำหนดรายละเอียดของโคบอลได้พิมพ์ออกมาเมื่อเดือนเมษายน 1960 และในสิ้นปีเดียวกันนั้นทางอาร์ซีเอและสเปอร์รี่ แรนด์ก็นำโคบอลคอมพายเลอร์ออกสู่ตลาด ซึ่งก็ตามมาด้วยผู้รายงานอื่น ๆ อย่างเร่งรีบ หน้าที่สำคัญอันหนึ่งของโคบอลก็คือ การจัดการในเรื่อง filing operations ต่าง ๆ สำหรับงานที่มีตัวเลขข้อมูลมาก ๆ แต่ก็มีผู้เห็นว่าเป็นภาษาที่ยังมีปัญหาหรือข้อเสียอยู่อีกหลาย

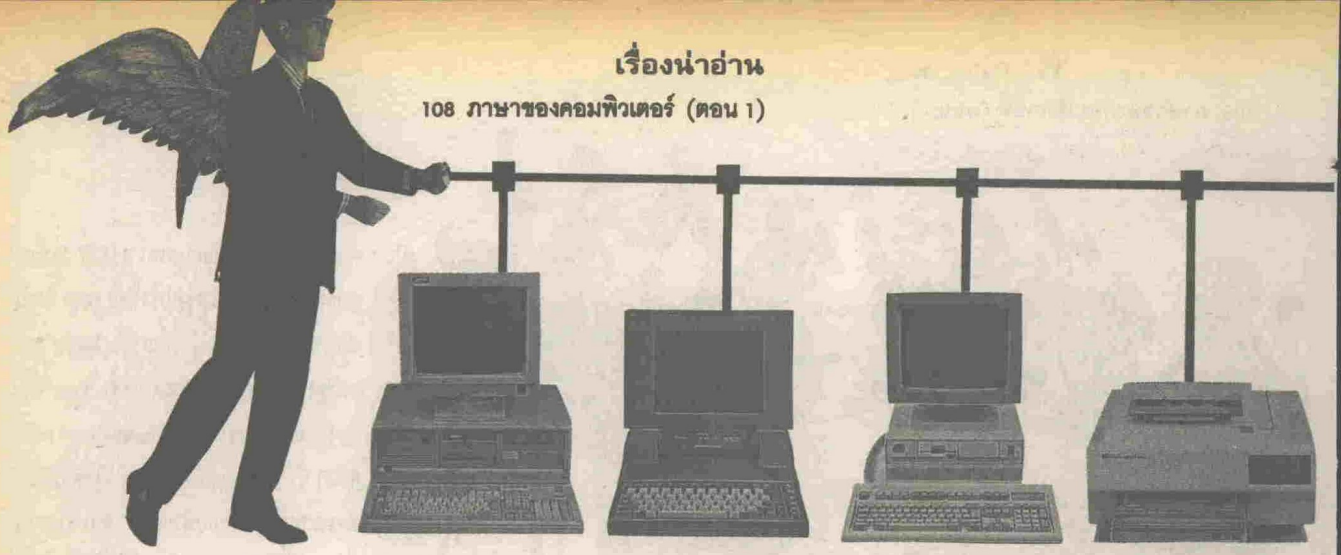
รูปที่ 8 คณะกรรมการผู้ออกแบบภาษา ALGOL 60 ที่มาพบกันในที่ประชุม ณ กรุงปารีสในปี 1960 โดยมีนักคอมพิวศาสตร์ชั้นนำ เช่น John Backus จาก สหรัฐฯ นักดาราศาสตร์ชาวดานิชชื่อ Peter Naur ฯลฯ จนเป็นที่มาของคำว่า "Backus-Naur" Form (BNF) ซึ่งเป็นระบบเครื่องหมายที่มีไว้พรรณนาลักษณะของภาษาคอมพิวเตอร์ ส่วนช่างภาพผู้ถ่ายรูปนี้ก็คือ John McCarthy ผู้ซึ่งในเวลาต่อมาได้เป็นคนสร้างภาษา Lisp ขึ้นมา

อย่าง เช่น เยิ่นเย้อเกินไป แม้ว่าจะมีผู้นิยมใช้เกือบค่อนโลกก็ตาม และทั้งภาษาฟอร์แทรนและโคบอลก็ยังไม่มีสิ่งใดที่ได้ชื่อว่าเป็นภาษาคอมพิวเตอร์สากล จนได้มีการสร้างภาษาอีกอันขึ้นมาในขณะที่โคบอลยังไม่เป็นตัวเป็นตนโดยเริ่มขึ้นที่เมืองซูริค (Zurich)

การคิดสร้างภาษาใหม่ที่พยายามจะให้เป็นภาษาสากลที่เรียกว่า ALGOL (Algorithmic Language) นี้เกิดจากความแพร่หลายของฟอร์แทรนนับตั้งแต่ออกสู่ตลาดเมื่อปี 1957 เป็นเหตุ เพราะทางยุโรปกลัวไอบีเอ็มจะครองโลก แต่ก็มีโปรแกรมเมอร์เป็นจำนวนมากในอเมริกาที่อยากได้ภาษาสากลสำหรับงานด้านวิทยาศาสตร์ที่ดีกว่าภาษาฟอร์แทรน ซึ่งเริ่มมีการขนานนามในขณะนั้นว่าเป็นเต่าล้านปี ความกระตือรือร้นของคนกลุ่มต่าง ๆ ในเรื่องนี้ได้ทำให้มีการพบปะในรูปของ international summit ขึ้นที่มหาวิทยาลัย Swiss Federal Technical ในเมืองซูริค เมื่อวันที่ 27 พฤษภาคม 1958 อันเป็นที่มาของคำว่า ALGOL 58 โดยมีนักคอมพิวเตอร์ศาสตร์เด่น ๆ ของโลกเข้าร่วมประชุมอย่างคับคั่ง ซึ่ง 8 คนจากจำนวนนี้ถือว่ามีชื่อที่สุดในขณะนั้นจัดได้ว่าเป็นแกนสำคัญของการพบปะ โดยมีชาวอเมริกัน 4 คน และเป็นชาวยุโรปอีก 4 คน

คติในการสร้างอัลกอลนั้นถือว่าการออกแบบภาษาคอมพิวเตอร์ก็เหมือนการสร้างวัตถุธรรมส่วนใหญ่ในแง่ที่ควรแข่งกับเวลา แม้ว่าผลิตผลอันเป็นภาษาที่ได้จะไม่สมบูรณ์นักก็ไม่เป็นไร จุดมุ่งหมายก็คือ จะทำให้อัลกอลมีความสามารถทางตรรกะที่หรูหรากว่าฟอร์แทรน แต่จนแล้วจนรอดการสร้างอัลกอลก็ยังต้องอาศัยอะไรหลาย ๆ อย่างจากฟอร์แทรนอยู่นั่นเอง ส่วนคำว่า ALGOL 60 ที่ปรากฏต่อมาในอีกระยะหนึ่งนั้นเกิดจากการประชุมที่กรุงปารีส เมื่อปี 1960 ดังแสดงในรูปที่ 8 จึงขอสรุปสั้น ๆ ว่าอัลกอลนั้นอุบัติขึ้นมาเหมือน ๆ กับโคบอล คือเกิดมาจากห้องประชุมที่เต็มไปด้วยควันบุหรี่ของคณะกรรมการที่มีทั้งการโต้เถียงดึงดันและการรอมชอมเป็นสรณะของการประชุมทำนองนี้ คือเหมือนกับโคบอลตรงจุดนี้ แต่ที่ต่างออกไปก็คือหลังจากที่ได้ปลุกปล้ำอยู่หลายปี คณะกรรมการอัลกอล ก็ได้มอบภาษาอันยั่วยวนใจไว้เป็นสมบัติของโลกมากกว่าประโยชน์ทางปฏิบัติ อันจัดเป็นภาษาคอมพิวเตอร์อันแรก ๆ ที่ได้ชื่อว่าหรูหราที่สุด จนดร.ฮ็อปเปอร์ได้ให้ความคิดเห็นไว้ในที่สุดว่า "ภาษาอัลกอลนั้นเปรียบเหมือนโคลงสี่ที่ยิ่งใหญ่ คือ เรียบง่ายและชัดเจน ในทัศนะของนักคำนวณ แต่ความยิ่งใหญ่เช่นนี้ไม่จำเป็นต้องมีประโยชน์เสมอไป"

ภาษาอัลกอลนี้ได้แพร่ไปในอเมริกาเช่นกัน แต่มีผู้นิยมน้อยมาก ด้วยเหตุผลหลาย ๆ อย่าง เช่น แม้คอมพิวเตอร์ที่ดีที่สุดในขณะนั้น (ปี 1959) ก็ยังไม่สามารถแปลความหมายของตัวอักขระทั้งหมดที่มี 116 ตัวซึ่งอยู่ใน reference-language character set ได้ ส่วนชาวยุโรปนั้นได้ตกหลุมรักอัลกอลอย่างจริงจัง เพราะเห็นว่าภาษานี้จะมาปลดแอกอุตสาหกรรมคอมพิวเตอร์ของภาคพื้นนี้ให้ไม่อยู่ใต้อาณัติของอเมริกา โดยได้มีการให้การสนับสนุนอัลกอลอย่างพร้อมใจกันในทุกประเทศในยุโรป นับตั้งแต่ German Research Council จนถึงหน่วยงานอื่น ๆ ที่เป็นของรัฐในหลาย ๆ ประเทศ อนึ่ง ภาษาอัลกอลนี้ได้เป็นต้นตอต่อไปอีกสำหรับหลาย ๆ ภาษา เพราะอัลกอลนั้นสร้างขึ้นมาด้วยความคิดอันเลิศเลอที่นำเอาความแปลกใหม่หลาย ๆ อย่างเข้ามารวมไว้ เช่น บล็อกสตรัคเจอร์ที่ยอมให้แบ่งโปรแกรมเป็นหน่วยย่อย ๆ ที่สมบูรณ์ในตัวเอง และสามารถเขียนหน่วยย่อย ๆ นี้แบบแยกกันชนิดต่างคนต่างเขียนได้โดยสะดวก ถัดมาก็เป็นจุดเด่นในเรื่องของ recursion อันเป็นคุณสมบัติที่โปรแกรมใดโปรแกรมหนึ่งจะสามารถอ้างอิงตัวของมันเองได้ คุณสมบัติอีกอันของอัลกอลก็คือเรื่อง



ของ BNF หรือ Backus-Naur Form ที่วางรากฐานในด้าน formal definition of syntax เอาไว้ที่จะเกี่ยวแก่การใช้คำต่าง ๆ ของภาษาให้เป็นไปอย่างแม่นยำ แนวความคิดขั้นพื้นฐานของภาษาอัลกอล ที่กล่าวมานี้เป็นต้นตอของภาษาอื่นดังได้กล่าวในตอนต้น

ภาษาในยุคของ System/360

ในปลายปี 1963 นั้นไอบีเอ็มกำลังมีแผนที่จะแนะนำคอมพิวเตอร์สำหรับใช้งานทั่วไป (general-purpose computers) รุ่นที่สำคัญที่สุดรุ่นหนึ่งออกสู่ตลาด ทำให้ต้องมองหาภาษาโปรแกรมสักอันที่น่าจะดีกว่าฟอร์แทรน โคบอล และอัลกอลรวมกัน เพื่อให้เหมาะกับคอมพิวเตอร์ใหม่สุดที่เรียกว่า System/360 ที่จะใช้ได้กับทั้งงานด้านวิทยาศาสตร์ และงานด้านประมวลผลข้อมูลของธุรกิจการค้าแบบต่าง ๆ รวมทั้งการใช้งานในลักษณะเฉพาะพิเศษ จึงได้ทุ่มเทความคิดพัฒนาภาษาอันหนึ่งขึ้นมา และได้เร่งรัดจะให้เสร็จทันการประกาศเปิดตัว System/360 ในเดือนเมษายนของปี 1964 ในรูปของ Programming Language One (PL/1) แต่ก็มีปัญหาในเรื่องของรายละเอียดต่าง ๆ ของภาษานี้ให้ต้องขบคิดและแก้ไขตกมากมายเหลือเกิน จนไม่มีทีท่าว่าจะเสร็จทันกำหนด ซึ่งในการพัฒนาภาษาที่เป็นส่วนหนึ่งของซอฟต์แวร์นี้จะต้องรอดูความคืบหน้าของการสร้างฮาร์ดแวร์ของ System/360 พร้อม ๆ กันไปด้วย

ในขณะนั้นได้คิดกันว่า ภาษาที่จะมาใช้คู่กับ System/360 อันเด่นดังนั้นจะต้องมีความสมบูรณ์ในหลาย ๆ ด้าน คือต้องมีอานุภาพสูง อันทำให้การสร้างภาษานี้เป็นงานใหญ่ที่ต้องการความเห็นจากหลาย ๆ ฝ่าย นับตั้งแต่สมาคมผู้ใช้เครื่องของไอบีเอ็มที่เรียกว่า SHARE ที่มีอิทธิพล จนถึงเอกชนผู้ใช้ที่เป็นรายใหญ่มาก ๆ อันทำให้ไอบีเอ็มต้องร่วมกับ SHARE เพื่อตั้งคณะกรรมการขึ้นมาชุดหนึ่งซึ่งเรียกว่า Advanced Language Development Committee โดยมีผู้แทนของ SHARE ที่มาจากล็อคฮีด, ยูเนี่ยนคาร์ไบด์ ฯลฯ รวมทั้งผู้เชี่ยวชาญของไอบีเอ็มเอง โดยเฉพาะนักสร้างคอมพายเลอร์มือฉมังของไอบีเอ็มผู้มีนามว่า George Radin ที่ได้รับการคัดเลือกให้เป็นหัวหน้าของโครงการนี้ โครงการสร้างภาษาใหม่นี้ได้มาเริ่มเอาเมื่อเดือนตุลาคม 1963 ส่วนการเปิดเผยตัวของ System/360 นั้นจะเป็นเดือนเมษายน 1964 ขณะเดียวกันผู้ที่อยู่ในแวดวงคอมพิวเตอร์ก็ต่างมองและหวังว่าภาษาหลักที่จะมากับ System/360 คงต้องเป็นภาษาฟอร์แทรน ซึ่งเป็นการคาดการณ์ที่ผิดหมด เพราะทางไอบีเอ็มไม่ได้คิดเช่นนี้แม้แต่น้อย เนื่องจากได้เห็นแล้วว่าฟอร์แทรนนั้นไม่ใช่ภาษาทั่วไป

System/360 ของไอบีเอ็มนั้นประกาศออกตัวเมื่อเดือนเมษายน 1964 ตามที่กำหนดไว้แต่แรกด้วยเหตุผลของการตลาด ส่วนการสร้าง PL/1 นั้นยังคงขับเคี่ยวกันไปด้วยการประชุมกันตามโรงแรมต่าง ๆ ทั้งในกรุงนิวยอร์กและลอสแองเจลิส โดยมีการอุทิศเวลาเพื่อเรื่องนี้เฉพาะวันหยุดสุดสัปดาห์เสียเป็นส่วนใหญ่ จนมาเสร็จเอาจริง ๆ เมื่อเดือนธันวาคม 1964 และเมื่อ PL/1 ปรากฏออกมาก็ได้รับการโจมตีจากผู้ใช้หลายกลุ่มว่าเป็นภาษาโปรแกรมที่ไม่มีรูปแบบที่ชัดเจน ตลอดจนมีความยุ่งยากในการใช้และเต็มไปด้วยความซับซ้อน บ้างก็โจมตีว่าเป็น มีดสวิสที่มีใบมีดถึง 100 ใบ สรุปแล้วคนส่วนใหญ่ไม่ค่อยชอบภาษานี้ในช่วงแรก ๆ จนกระทั่งได้มีการปรับปรุงต่อไปอีกพักหนึ่ง จึงเริ่มยอมรับ ภาษา PL/1 ก็เหมือนภาษาคอมพิวเตอร์ส่วนใหญ่ที่เป็นผลมาจากรากฐานที่วางไว้ไม่ดีในตอนแรก ๆ ของยุคคอมพิวเตอร์ ซึ่งในเวลาต่อมาก็มีแต่การขอยืมความคิดจากภาษารุ่นก่อนมาใช้จนไม่มีการมองหาทางออกใหม่ ๆ มาช่วยจรรโลงให้ภาษาเช่นนี้รุ่งโรจน์โดดเด่นในแง่ของการใช้งาน

เราจะพูดถึงยุคของการตื่นตัวของภาษา, โลกของ AI, ยุคของ time sharing ใน 108 ภาษาของคอมพิวเตอร์ ตอน 2 พบกันใหม่ในฉบับหน้าครับ